

RS232 Comms port on Casa controller

Paul Dorrell, Malcolm Law & Richard Dudley, B&W Loudspeakers SRE

Document revision I, 16 Oct 2000

Scope : Software revision 2.14 or later

Physical

9600 baud serial, 8 bits, no parity, 1 start bit, 1 stop bit.

No flow control - RTS/CTS tied together, no Xon/Xoff protocol.

9 pin D-type female connector, receive on pin3, transmit on pin2

Note : in installations with multiple networked controllers, each will need to be controlled over its own RS232 port. There is no provision in the protocol for accessing other networked controllers via RS232.

Flash EPROM updating

In order to update the on-board Flash memory, the CPU must be set into boot mode. This is done by setting Config switch 2 in the down position, with the power off, and then powering up. The DOS program 'bloder.exe' is provided to download revised software. The CPU also has another form of non-volatile storage in the form of battery-backed RAM. We strongly recommend that this is also purged after a software upgrade – this can be achieved by means of the Visual Basic program provided. Further instructions regarding downloading new software are provided with the boot loader and any revised software.

Note on controllers with serial numbers below 600, the config 2 switch is inoperable. Boot mode is engaged with a programming link on the CPU card. To access the CPU card, remove six screws in the underside of the lid and three along the top of the rear panel.

Operational communications

A bi-directional command and status channel is provided between the Casa controller and any external equipment according to the following definitions.

Structure

Data (in both directions) is transferred in 'packets'.

A packet is a sequence of 2-17 bytes with the following format

Byte 1 : Most significant nibble checksum, least significant nibble n = number of data bytes

Byte 2 : Type of packet. See list of valid packet types below.

Byte 3 to 2+n : Extra data.

The checksum is such that the binary sum of the nibbles is equal to 0xC modulo 0x10.

Types of Packet

0 Error. Byte 3 contains an error number, subsequent bytes may give additional info.

- 1 Is anybody out there? Just acknowledge if you hear me.
- 2 Set Linkages.
Used to define what information the Casa system transmits on the RS232 (default none), and to define if any internal data links (eg passing keypad keypresses though to user interface) are to be broken. The default condition is that all internal data links are intact. See below for further description of linkages.
- 3 Keypress.
4 extra bytes : Zone, Keyno, Age of keypress (2 bytes low, high).
Received from Casa signifies a key is pressed on the keypad.
Transmitted to Casa 'pokes' a keypress into the user interface.
An 'age' of -1 indicates 'key released' and is not essential.
A key may repeat (whilst held down), with strictly increasing ages which do not have to be consecutive. There should be around 20 'ages' to the second, and a 'long keypress' is deemed to be one with an 'age' of > 3 seconds. See appendix A1 for tabulation of key values.
- 4 Display.
Zone, 8 characters text.
Appendix B points to the Siemens datasheet which depicts the character set available, using 7-bit characters.
The MSB of each character can be set to indicate flashing.
- 5 Preamplifier hardware.
See below for additional packet structure.
- 6 Preamplifier user interface.
Zone no 0-3 (corresponding with zones A-D respectively)
Zone no.s 128-131 (i.e. MSB set) denote an enquiry and require the function only.
Function Bit 0=volume, Bit 1=source, Bit 2=balance, Bit 3=bass, Bit 4=treble, Bit 5=Preset local input volume
Values signed bytes. The specified values in order volume,...,treble.

Range for bal,bass,treb is [-7,+7] , +ve balance is 'panned right', or left channel attenuated.
Source is 0-3 for controller sources, 4 for local input.
The Preset local i/p volume is for applications where a fixed volume setting is required for the local i/p (the value given will be rounded to a valid level currently 10,15,...,80). If the normal variable volume level is required, it should be set to 0.

Example
0x84,0x06, 0x02,0x09,0x30,0xFD
Set the volume on zone 2 to 48, and the bass on zone 2 to -3
- 7 Erase Srctable in Flash - to amend supported source equipment
This function is not currently supported - so do not use.
- 8 Write Srctable Flash - for updating supported source equipment
This function is not currently supported - so do not use.

9 Configure Source

Source no. 0-3 (With bit 7 set denotes an enquiry : no further bytes needed)

Global id 0-15

Casa id 16 bits : low, high.

(0 if equipment is controlled by another Casa controller in the network)

The Global id identifies a particular item of equipment that is controlled by the network of Casa systems, whilst the Casa id is a reference into the Casa database of types of source equipment.

Example packet

0xD3,0x09, 0x00, 0x02, 0x01.

This says that the audio coming into this controller on channel 0 is from equipment 2. Equipment 2 is controlled by this controller (not any others), and is of type 1 (a Rotel Tuner).

There is a further complication in that equipment of global id 0xC-0xF (if present) must be locally controlled. If two separate controllers reference equipment 0xC they mean different pieces of equipment.

10 Set time

Hours (0-23)

Minutes (0-59)

Seconds (0-59)

Sets the time on the whole network, not just this controller. See appendix A3 for an example.

11 Alarm

Zone no 0-3

Zone no.s 128-131 (i.e. MSB set) denote an enquiry and need no further bytes.

enabled Bit 0=Alarm on enabled, 1=Alarm off enabled, 2=Clock display enabled

Minutes on 0-59

Hour on 0-23

Source on 0-3

Minutes off 0-59

Hour off 0-23

See appendix A2 for an example.

Error Codes

0 No Error : Acknowledgement.

Subsequent bytes (if present) are

Major version no.

Minor version number (if top bit set it is interim - should not be seen outside B&W).

Source Major version no.

Source Minor version number (top bit set is interim).

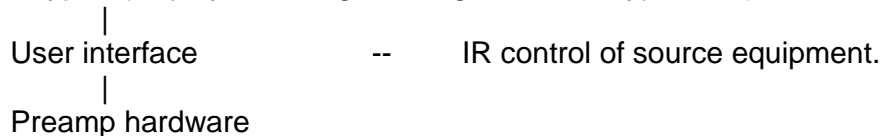
1 Timeout.

- 2 Checksum Error.
- 3 Bad command.
- 4 Can't do (eg attempt to erase flash without enabling linkage).
- 5 Flash not erased (when writing an area).
- 6 Casa Id not known (source set to something not defined in Flash)

Set Linkages

The principle here is that internally, the Casa control software is composed of several communicating subsystems.

Keypad (displays messages and generates keypresses)



Any of these communications links can have information 'poked' into them which will be understood as if it came from the legitimate source. Further, any of these links can be set-up to output information to the RS232 interface, or 'peeked'. The direct links can also be completely severed and perhaps replaced with external connections in the driving software, thus completely changing the 'personality' of Casa. Be warned though that the responsiveness of the whole system will then depend on any latencies introduced by the controlling system.

It should also be noted that the RS232 communications are to a particular casa controller - if a network of them is required to be controlled this can only be accomplished by dealing with each controller individually over its RS232 port.

The user interface to IR control link cannot be hooked into as of yet.

Data format

Byte 3 : Type of linkage

Byte 4 : Enable/disable mask.

Linkage types :

1 Display information

2 Keypresses

3 Preamp

Mask :

Zones A-D correspond to bits 0-3 and 4-7 respectively.

Bits 0 through 3 are set for 'want to peek that zone'.

Bits 4 through 7 are set for 'cut internal link on that zone'

Preamplifier hardware

For flexibility, the link between the user interface and the preamp hardware is defined to be at quite a low level - at the level of the control bytes sent over an internal serial interface to the TDA7318 preamp chip. Moreover it is quite feasible

Packet type = 0x03
Zone B = 0x01
Vol+ = 0x0F
Age = 0x0000

Example - to select Tuner on zone D send the following six bytes

0x04, 0x03 , 0x03, 0x02, 0x00, 0x00

Composed of :
Checksum = 0x0
Number of data bytes = 0x4
Packet type = 0x03
Zone D = 0x03
Tuner = 0x02
Age = 0x0000

Example - a long (3 second) keypress on CASA to set timer on zone B send the following six bytes

0x04, 0x03 , 0x01, 0x11, 0x3F, 0x00

Composed of :
Checksum = 0x0
Number of data bytes = 0x4
Packet type = 0x03
Zone B = 0x01
Casa = 0x11
Age = 0x003F

Example - a very long (6 second) keypress on CASA to set clock send the following six bytes

0x64, 0x03 , 0x01, 0x11, 0x75, 0x00

Composed of :
Checksum = 0x6
Number of data bytes = 0x4
Packet type = 0x03
Zone B = 0x01 NB: this identifies the zone keypad, however the clock, once set, is common to all zones.
Casa = 0x11
Age = 0x0075

The above keycodes are also used for keypads assigned to SubzoneA in a zone configured for subzone operation. To generate keycodes for other subzones these basic keycodes need to be added to 32, 64, and 96 for Subzones B, C, and D respectively. For example, the keycode for "On/Off" from SubzoneB is $(21 + 32 = 53)$ 0x35. Also, the keycode for "Vol+" from SubzoneC is $(15 + 64 = 79)$ 0x4F.

Appendix A2 : setting alarm

This example uses packet type 11, high level command for setting an alarm, and sends the following nine bytes

0x57, 0x0B, 0x01, 0x04, 0x00, 0x08, 0x00, 0x00, 0x17

Composed of :

Checksum = 0x5

Number of data bytes = 0x7

Packet type = 0x0B

Zone B = 0x01

Enable = 0x04 enable clock, disable on time, disable off time.

Minute on = 0x00

Hour on = 0x08 on at 8:00 am

Source = 0x00 source 1 (tuner)

Minute off= 0x00

Hour off= 0x17 off at 11:00 pm

The hour parameter represents "hour of day" in a 24 hour format.

Appendix A3 : setting clock time

This example uses packet type 10, high level command for setting clock time, and sends the following five bytes

0xF3, 0x0A, 0x0F, 0x25, 0x19

Composed of :

Checksum = 0xF

Number of data bytes = 0x3

Packet type = 0x0A

Hour = 0x0F Decimal 15 which is 24 hour format for 3 pm.

Minute = 0x25

Second = 0x19 3:37:25 pm

Appendix A4 : calculating checksums

The very first "nibble" (half a byte) of any string sent is call the "checksum correction nibble". So in the example string:

0x54, 0x03, 0x01, 0x0F, 0x00, 0x00.

this has a value of 5. This nibble is calculated so that the sum of all the nibbles in the string equals hexadecimal C. So for the above string:

$5 + 4 + 0 + 3 + 0 + 1 + 0 + F + 0 + 0 + 0 + 0 = 0x1C.$

This sum modulo 0x10 (taking the last hexadecimal digit) should always equals hex C. So the "checksum correction nibble" is simply a means of ensuring that the real checksum always equals C. At the receiving end, if the calculated checksum is not C then the string is rejected.

Of course, when composing the string, the user will need to perform the calculation in reverse. So, in the context of the example, this would be as follows:

$$\begin{aligned}\text{Checksum correction nibble} &= 0x0C - (4 + 0 + 3 + 0 + 1 + 0 + F + 0 + 0 + 0 + 0) \\ &= 0x0C - 0x17 \\ &= 0xF5 \\ &= 0x5 \text{ (modulo } 0x10\text{)}\end{aligned}$$

Appendix B

(Siemens datasheet for DLG2416)

<http://www.sci.siemens.com/pdf/opto/intelligent/dlr2416.pdf>

Appendix C

(STM datasheet for TDA7318)

<http://www.st.com/stonline/books/pdf/docs/1491.pdf>

Revision History

Revision C 16 Dec 97

Optional version number added to Acknowledge.
Examples made into complete packets.

Revision D 10 Feb 98

Local input Preset volume added to Configure Source

Revision E 17 April 98

Acknowledge response extended to include source versioning

Revision F,G July/Sept 98

Further example packet added, keycodes identified

Revision H Jan 99

Code for local input present included, hyperlinks added to pdf files for appendices.

Revision I Oct 2000

Extensions to Appendix A1 to cover keycodes from small keypads in Subzones B, C, and D.